Application Note AN-103
InnoSwitch4-Pro Family

PIC Library Overview and Guide

Introduction

The application of InnoSwitch-Pro Family PIC Library is discussed in this document. This library can be used with both the InnoSwitch3-Pro and InnoSwitch4-Pro family of devices. This code was designed to be highly portable with different microcontroller platforms. The use of C++ language will make it easy for users to understand and modify the code according to their needs. This guide will allow the user to get sufficient knowledge on how to operate the devices with a use of a widely used microcontroller such as PICs.

InnoSwitch4-Pro

InnoSwitch4-Pro devices are ideal for AC/DC power supply applications where fine (10 mV, 50 mA) output voltage and current adjustment are necessary. Typical implementations comprise a system microprocessor or dedicated microcontroller with an I^2C port that is used to configure, control and supervise the operation of the power sub-system. The uVCC pin provides a bias supply for the microprocessor in stand-alone implementations such as USB PD adapters and chargers.

The command and telemetry registers on InnoSwitch4-Pro are updated compared to InnoSwitch3-Pro. These features add flexibility and improve fault response. An example of these changes is with the Constant Current register resolution which is increased from 128LSB to 192 LSB and therefore provides better control on the output current.

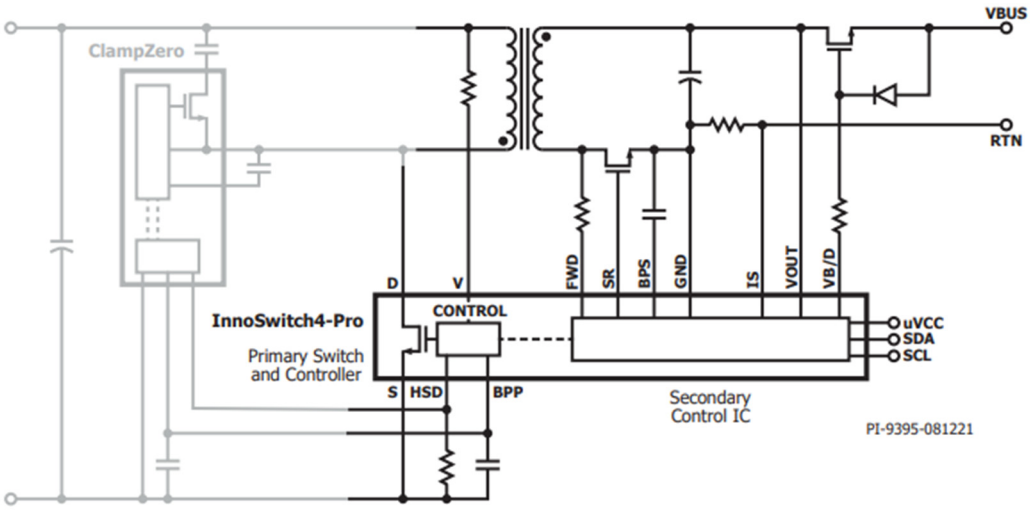


Figure 1. InnoSwitch4-Pro Schematic

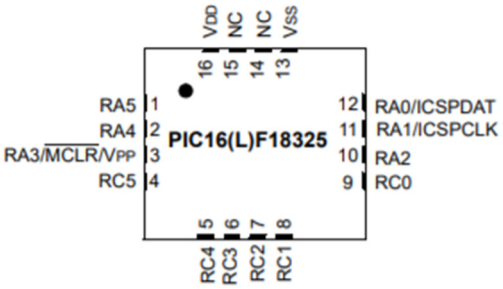


Figure 2. PIC16F18325 16-Pin UQFN Diagram

This demo runs on a 65W Reference Design Board (DER-961) with a PIC device as the microcontroller. The PIC microcontroller used in this design is the 16-pin UQFN packaged PIC16F18325. J1 is configured as the programming header based on the PICkit 4 Pin-out (Figure 5). Header pins J2, J4, and J6 are shorted to bridge the I²C and uVcc lines of the InnoSwitch4-Pro to the PIC microcontroller. By removing the short on J2, J4, and J6 pins, an external I²C interface header can be used to bypass the I²C signals of the PIC microcontroller to the InnoSwitch4-Pro.

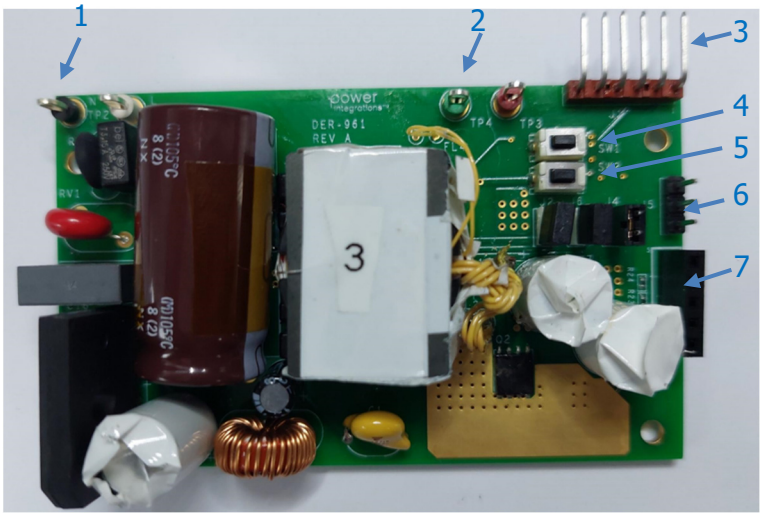


Figure 3. DER 961 Front

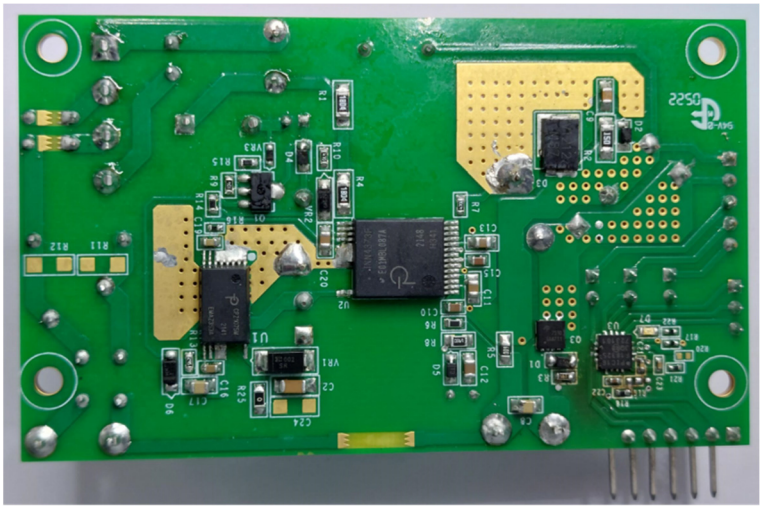


Figure 4. DER 961 Back

Number	Description	Label
1	AC Input Terminals	TP1, TP2
2	DC Output Terminals	TP3, TP4
3	PICkit4 Programming Header	J1
4	MCU Push Buttons	SW1, SW2
5	uVcc and I²C Isolation Headers	J2, J4, J6
6	MCU GPIO Headers	J5
7	External I²C Interface Header	J3

Table 1. DER 961 Part Description

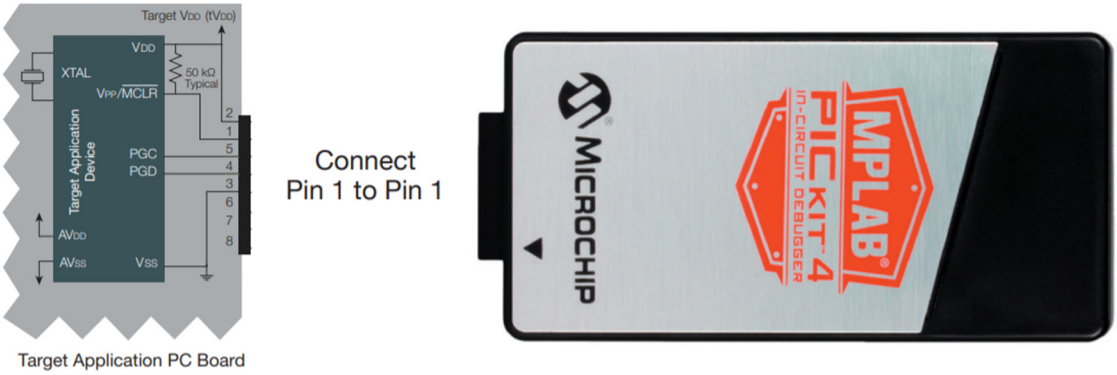


Figure 5. PICkit Programming Header Pin Mapping

Code Library

The InnoSwitch4-Pro Family library consists of both InnoSwitch3-Pro and InnoSwitch4-Pro drivers and examples. The combined library makes it easy to switch from one InnoSwitch-Pro device to another. Click on the link below to go to the Power Integration's website and download the InnoSwitch3-Pro/InnoSwitch4-Pro PIC Library.

https://www.power.com/

Software Settings

MPLAB IDE is the software used to develop the InnoSwitch3-Pro/InnoSwitch4-Pro PIC library for its ease of use. The IDE has a variety of tools specifically designed for PIC microcontrollers and embedded systems. The setting below ensure that the PIC library will not encounter any compatibility issues while compiling and uploading.

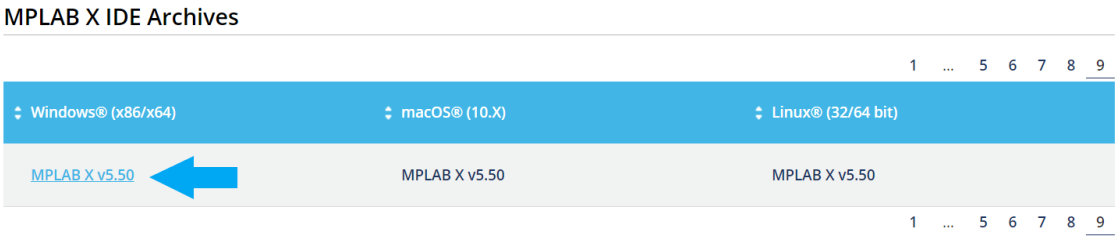
MPLAB X v5.50

- MPLAB Code Configurator v5.0.3
- PIC16F1xxxx_DFP v1.8.149
- XC8 v2.32 Compiler

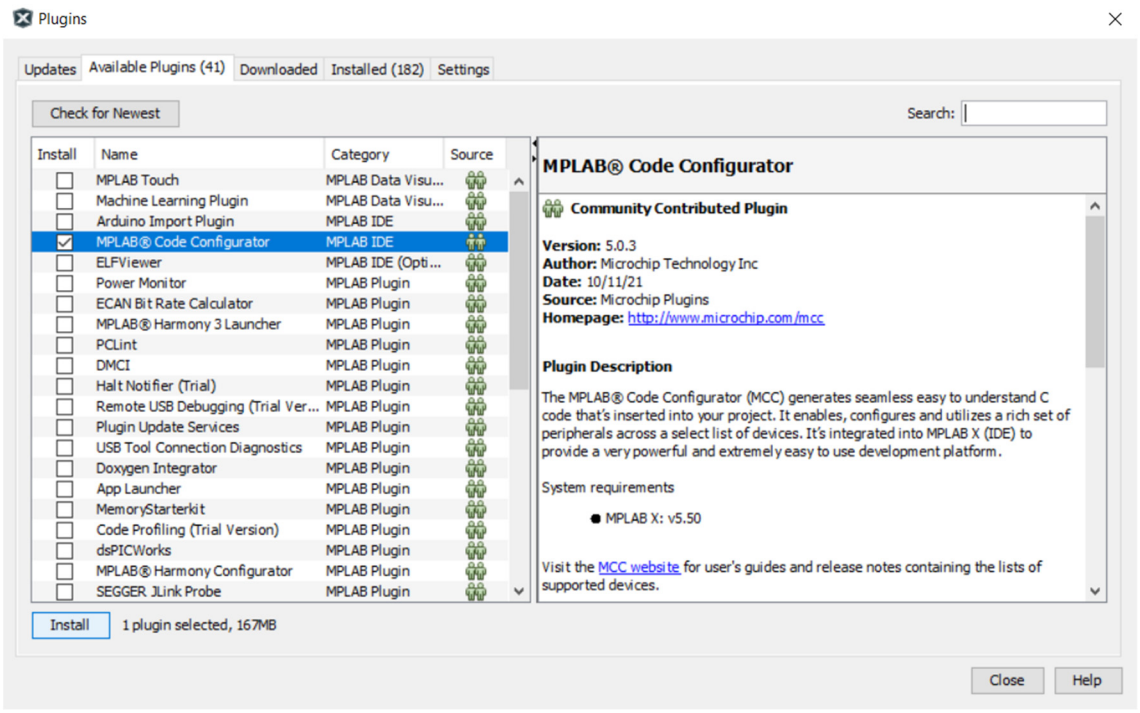
Installing MPLAB IDE

Download the XC8 installer from the Microchip website below. The XC8 installer can be found under the MPLAB X IDE Archives.

https://www.microchip.com/en-us/tools-resources/archives/mplab-ecosystem

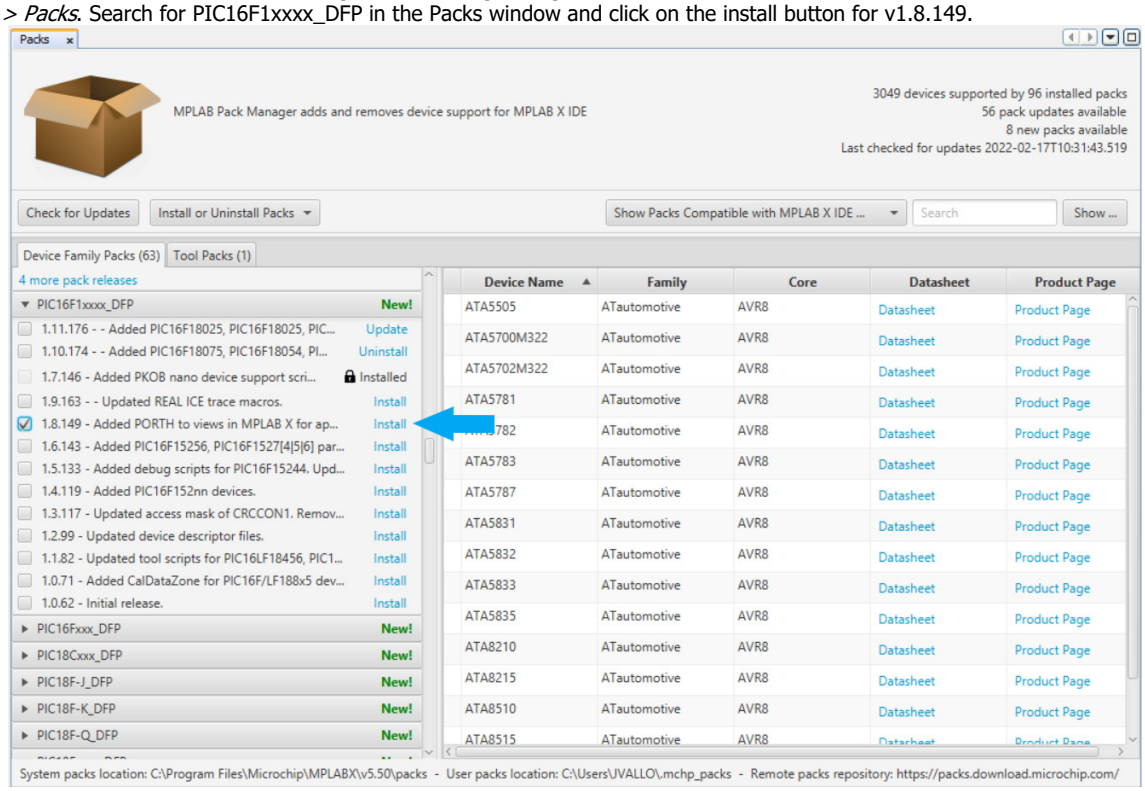


In the MPLAB X IDE. Go to *Tools > Plugins Download* and search for the MPLAB Code Configurator under the Available Plugins tab and click install.



Installing the PIC16F1xxxx_DFP v1.8.149(Device Family Pack)

Go to *Tools* > *Packs*. Search for PIC16F1xxxx_DFP in the Packs window and click on the install button for v1.8.149.



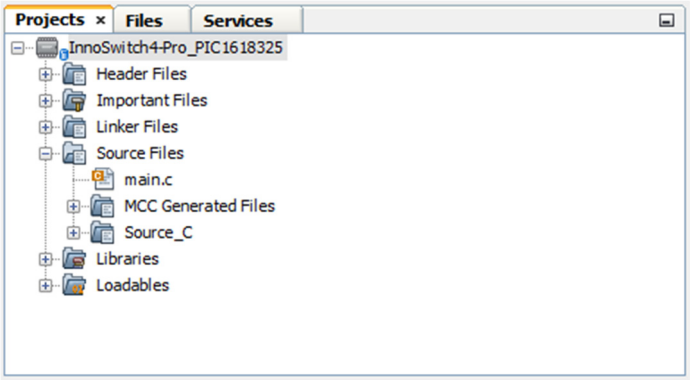
Installing XC8 v2.32 Compiler

Download the XC8 installer from the Microchip website below. The XC8 installer can be found under the Language Tool Archives.

Driver and Source Files

Open the InnoSwitch4-Pro Library in the MPLAB IDE. Download and extract the InnoSwitch4-Pro PIC Library from the link below.

https://www.power.com/



The driver files serve as the core of the InnoSwitch4-Pro library. The files handle timings, communication, and InnoSwitch4-Pro registers. InnoSwitch3-Pro drivers are also included in the library. However, they are disabled by default.

- Source_C
  - app
    - app.c
    - buttons.c
    - led.c
  - device
    - inno3Pro
      - inno3Pro.c
    - inno4Pro
      - inno4Pro.c
    - innoProBase
      - innoProBase.c
  - hal
    - hal_i2c.c
    - hal_timer.c
  - main_examples
    - InnoSwitch3-Pro
    - InnoSwitch4-Pro

App – Handles application routines specifically for functions related to PIC GPIO.

Device – Handles command sequences, timings, register settings, threshold calculations, parity implementations, telemetry, and etc. related to InnoSwitch4-Pro registers.

Hal – Also known as the hardware abstraction layer. Files under this section manages the I^2C packet format based on InnoSwitch4-Pro datasheet. In addition, they control the hardware timings and clock signals of the PIC microcontroller.

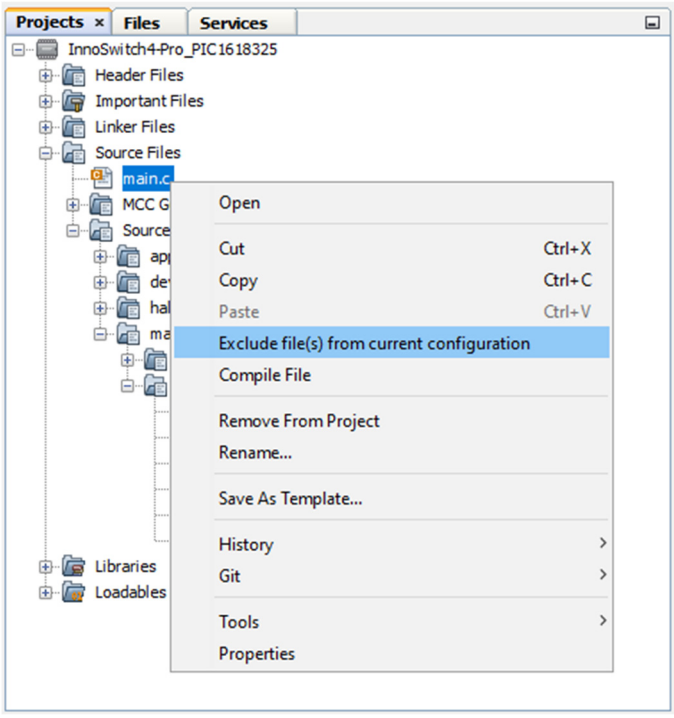
NOTE: The files inno3Pro.h, inno3Pro_Config.h, and inno3Pro.c is excluded from the compilation by default.

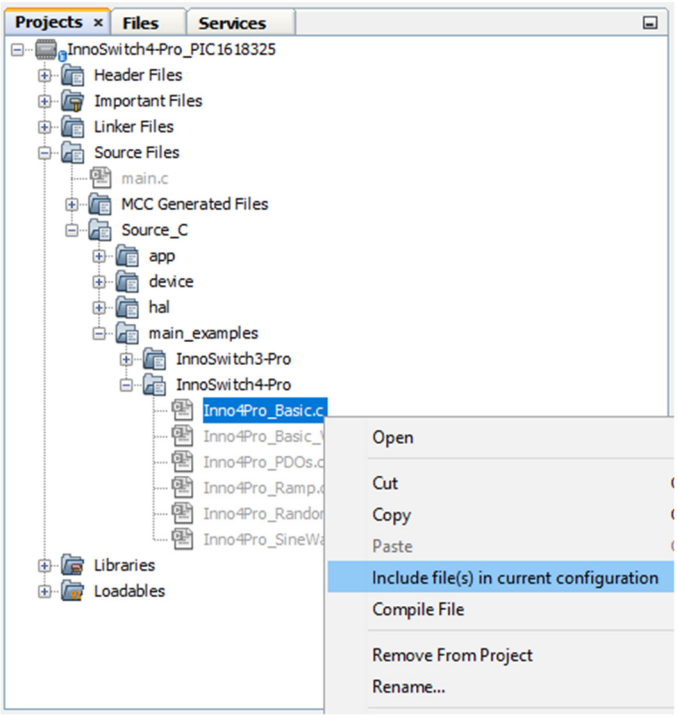
Examples

The example implementations can be found in *Source Files > Source_C > main_examples.* These files contain implementations for different use cases of the InnoSwitch4-Pro register. The examples serve as a guide or template for the user on different functions such as initializing the InnoSwitch4-Pro device and changing the output voltage.

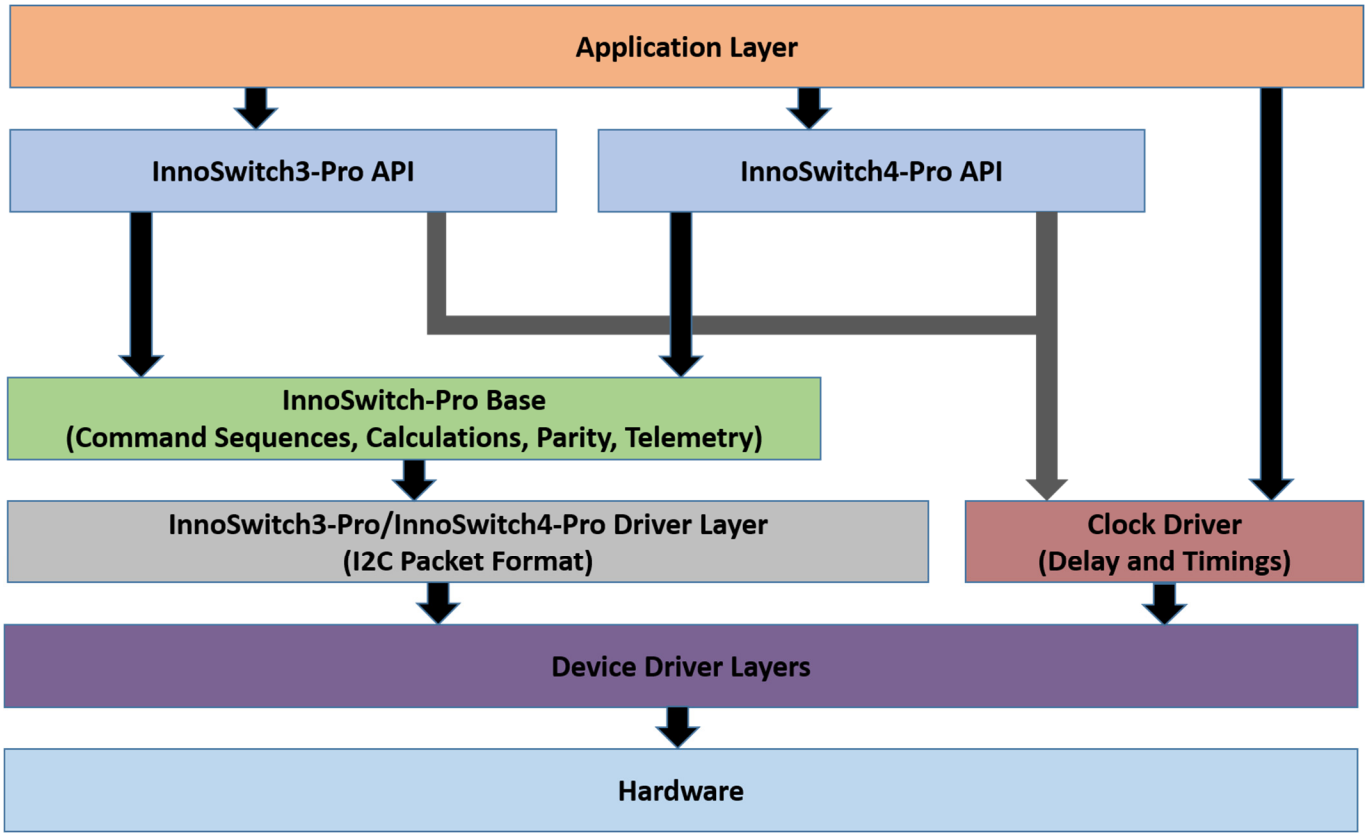
- Source_C
  - app
  - device
  - hal
  - main_examples
    - InnoSwitch3-Pro
      - Inno3Pro_Basic.c
      - Inno3Pro_Basic_Volt_Amps_OV_UV.c
      - Inno3Pro_PDOs.c
      - Inno3Pro_Ramp.c
      - Inno3Pro_Random_Volt_Timer.c
      - Inno3Pro_SineWave.c
    - InnoSwitch4-Pro
      - Inno4Pro_Basic.c
      - Inno4Pro_Basic_Volt_Amps_OV_UV.c
      - Inno4Pro_PDOs.c
      - Inno4Pro_Ramp.c
      - Inno4Pro_Random_Volt_Timer.c
      - Inno4Pro_SineWave.c

The files under the main_examples folder are excluded from the configuration by default. The compiler will ignore the files excluded from the current configuration. In order to use an example file, the user must exclude the **main.c** file and include the example file, in this case, **Inno4Pro_Basic.c**. This is done through a right click on the file and selecting **Exclude/Include files(s) from current configuration**.





The figure below shows how each of the layers interact with each other. The application layer consists of the main.c and main_example files that implement the function of the InnoSwich4-Pro and Clock Driver functions. InnoSwitch3-Pro and InnoSwitch4-Pro API contain the command sequences, calculations, parity, and telemetry functions specific to them. InnoSwitch-Pro base contains the functions similar to the two API.



Example 1 – Inno4Pro_Basic.c

Inno3Pro_Basic.c file implements the essential commands to run the InnoSwich3-Pro IC. The adapter will output 5V 3.1A while using this code. There are five commands in this file to note:

Inno4Pro_Initialization();	Function for initializing the InnoSwitch4-Pro device
Inno4Pro_Write_VI();	Function for setting the output voltage and current
Inno4Pro_Write_Volt_Peak();	Sets the knee voltage (V_{KP})
Inno4Pro_Vbus_Switch_Control();	Controls the BUS switch

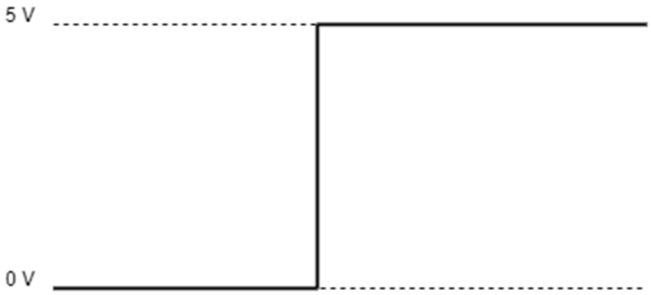


Figure 8. Inno4Pro_Basic.c Output Voltage Waveform

```
//MPLAB Code Configurator Header File
#include "../../../mcc_generated_files/mcc.h"

//Step  1  :  Add the Header Files
#include "../../../src/hal/hal_i2c.h"
#include "../../../src/hal/hal_timer.h"

#include "../../../src/inno4Pro/inno4Pro_Config.h"
#include "../../../src/inno4Pro/inno4Pro.h"


void main(void)
{

    // Initialize the device - PIC16F18325
    SYSTEM_Initialize();
    INTERRUPT_GlobalInterruptEnable();
    INTERRUPT_PeripheralInterruptEnable();


    //Step 2 :  Write Initial Commands to InnoSwitch4-Pro
    Inno4Pro_Initialization();


         //Step 3 :  Call the Functions on the Main Loop
    while (1)
    {
      // Main Loop Variable Initialization
      float fVolts = 5;                       //Initialize Voltage at 5V
      float fAmps = 3.1;                      //Initialize Constant Current at 3.1A
      float fCableDropComp = 300;             //Initialize Cable Drop Compensation to 300mv
      float fVoltPeak = 24;                   //Initialize Knee Voltage at 24V
      float fVbusEn = 1;                      //Initialize Vbus Enable to at ON

      //Library Call in the Mainloop
      Inno4Pro_PD_Write_VI            ( fVolts  , fAmps   );     //Set Voltage and current
      Inno4Pro_Write_Cable_Drop_Comp  ( fCableDropComp    );     //Set Cable Drop Compensation
      Inno4Pro_Write_Volt_Peak        ( fVoltPeak         );     //Set Constant Output Power Knee Voltage
      Inno4Pro_Vbus_Switch_Control    ( fVbusEn           );     //Set Vbus Enable

      }
}
```

Example 2 – Inno4Pro_PDOs.c
This file cycles through the multiple output voltages. This mimics the USBPD standard PDO's for 60W adapters.

Inno4Pro_Initialization();	Function for initializing the InnoSwitch4-Pro device
Inno4Pro_Write_VI();	Function for setting the output voltage and current
Inno4Pro_Write_Volt_Peak();	Sets the knee voltage (V_{KP})
Inno4Pro_Vbus_Switch_Control();	Controls the BUS switch
clock_HasTimeElapsedMs();	Returns 1 when a certain amount of time in milliseconds has elapsed
Clock_GetTimeStapsMs();	Returns the current time in milliseconds
Inno4Pro_PD_Write_VI();	Another implementation of Inno4Pro_Write_VI() specifically for USBPD applications

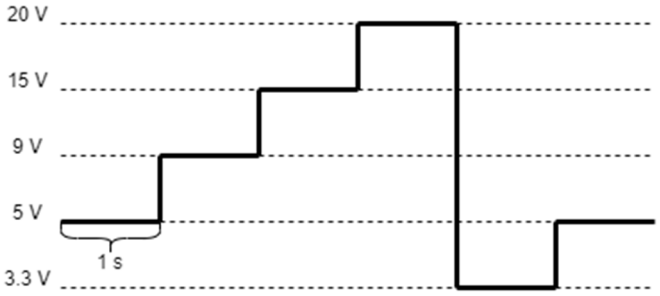
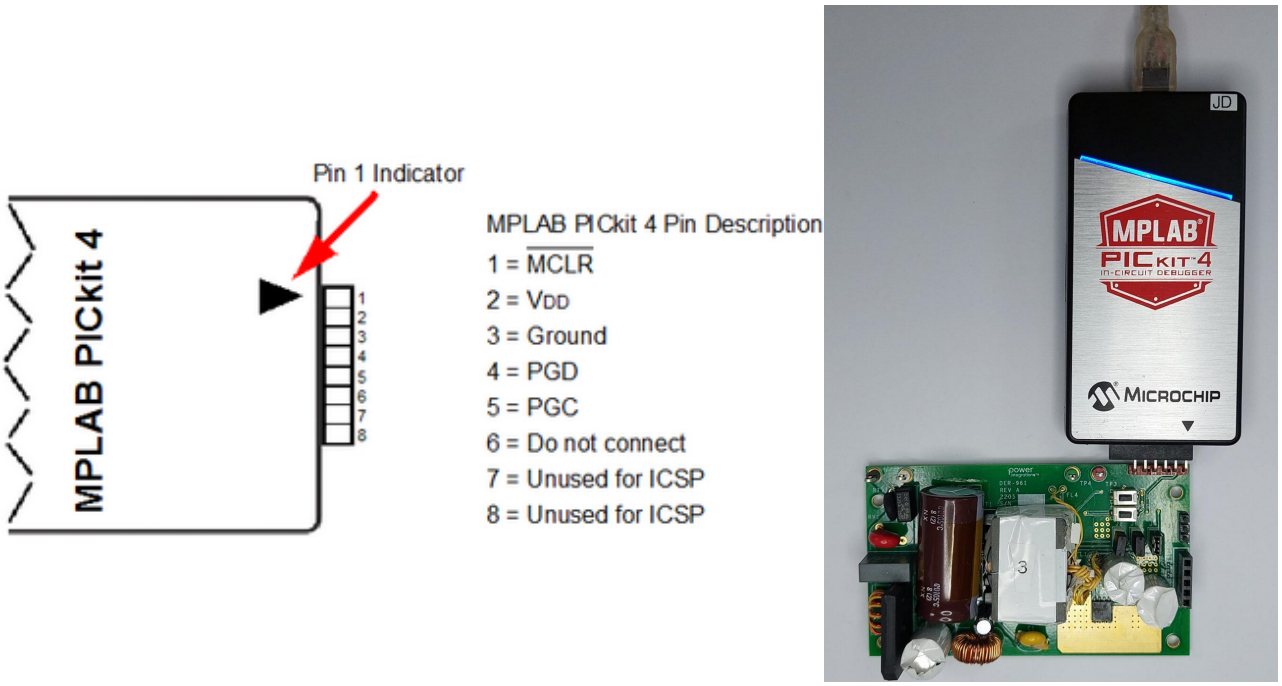


Figure 9. Inno4Pro_PDOs.c Output Voltage Waveform

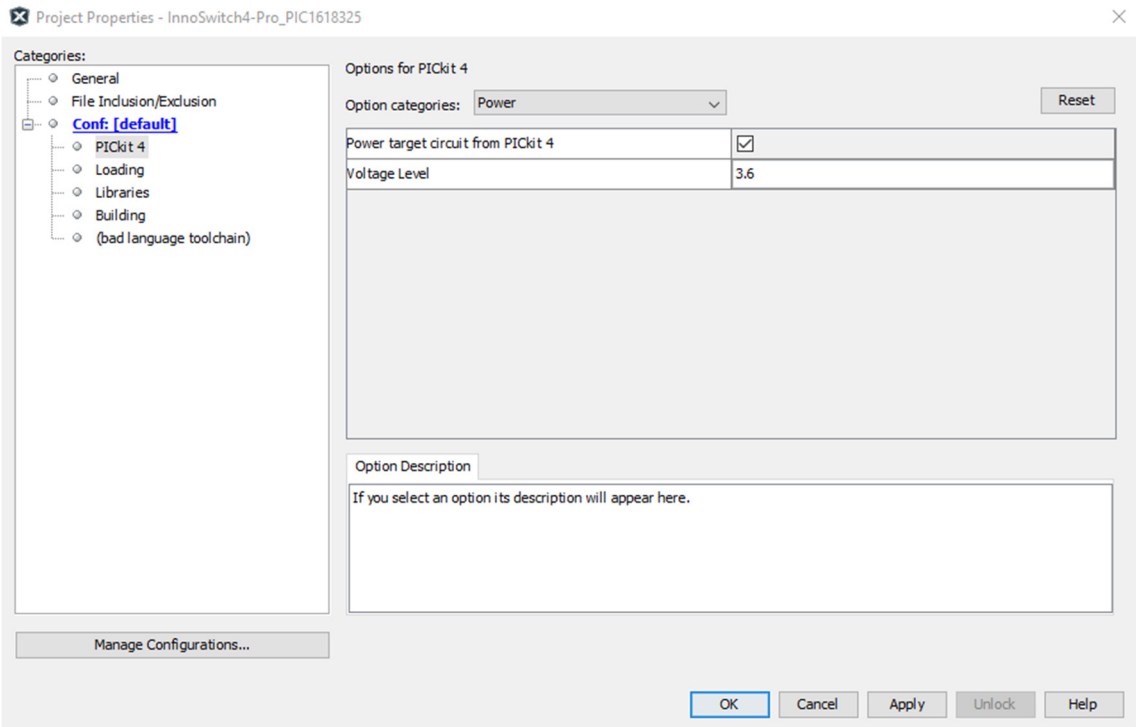
```
//MPLAB Code Configurator Header File
#include "../../../mcc_generated_files/mcc.h"

//Step  1   :  Add the Header Files
#include "../../../src/hal/hal_i2c.h"
#include "../../../src/hal/hal_timer.h"
#include "../../../src/hal/hal_gpio.h"

#include "../../../src/inno4Pro/inno4Pro_Config.h"
#include "../../../src/inno4Pro/inno4Pro.h"
#include "../../../src/app/buttons.h"

void main(void)
{
    // Initialize the device - PIC16F18325
    SYSTEM_Initialize();
    INTERRUPT_GlobalInterruptEnable();
    INTERRUPT_PeripheralInterruptEnable();

    //Write Initial Commands to InnoSwitch4-Pro
    Inno4Pro_Initialization();
    Inno4Pro_Write_Cable_Drop_Comp(300);      // CDC  = 300mV
    Inno4Pro_Write_Volt_Peak(24);             // VKP  = 24V
    Inno4Pro_Vbus_Switch_Control(1);          // VBEN = ON
    //Call the Functions on the Main Loop
    while (1)
    {
        // Main Loop Variables
        static uint16_t u16_Main_State = 0;                //Main State Counter

        // Switch Debounce
        if(debounce(Button1_DigitalRead(),1))
        {
            u16_Main_State++;
        }
         switch(u16_Main_State)
         {
         case 0:
                  u16_Main_State = 1;
                  Inno4Pro_Write_Cable_Drop_Comp(300);      // CDC  = 300mV
                  Inno4Pro_Write_Volt_Peak(24);             // VKP  = 24V
                  Inno4Pro_Vbus_Switch_Control(1);          // VBEN = ON
                  break;

         case 1:   //Activate 5V Configuration
                  Inno4Pro_PD_Write_VI(5,3.1);              // CV = 5V and CC = 3.1A
                  break;

         case 2:     //Activate 9V Configuration
                  Inno4Pro_PD_Write_VI(9,3.1);
                  break;

         case 3:   //Activate 15V Configuration
                  Inno4Pro_PD_Write_VI(15,3.1);
                  break;

         case 4:     //Activate 20V Configuration
                  Inno4Pro_PD_Write_VI(20,3.1);
                  break;

         case 5:   //Activate 3.3V Configuration
                  Inno4Pro_Write_VI(3.3,3.1);
                  break;

         default:
                  u16_Main_State = 1;
                  break;
         }
```

Programming

Connect the PICkit 4 to the programming header of the PiC device. The picture below shows the correct connection of the PIC IC pins to the PICkit 4 programmer.



Note: When programming the PIC device without the microcontroller voltage supply. Check the **Power target circuit from PICkit4** option in the PICkit 4 properties. Change the voltage level to 3.6 to 3.7 V to power up the PIC microcontroller. Uncheck the **Power target circuit from PICkit4** when programming a PIC device that is powered up.



Open the project properties from the project dashboard located beneath the projects window. Select PICkit 4 as the Connected Hardware Tool, 1.8.149 under the Packs pane, and XC8 v2.32 under the Compiler Toolchain. Make sure that the proper PIC device is selected which in this case is PIC16F18325.

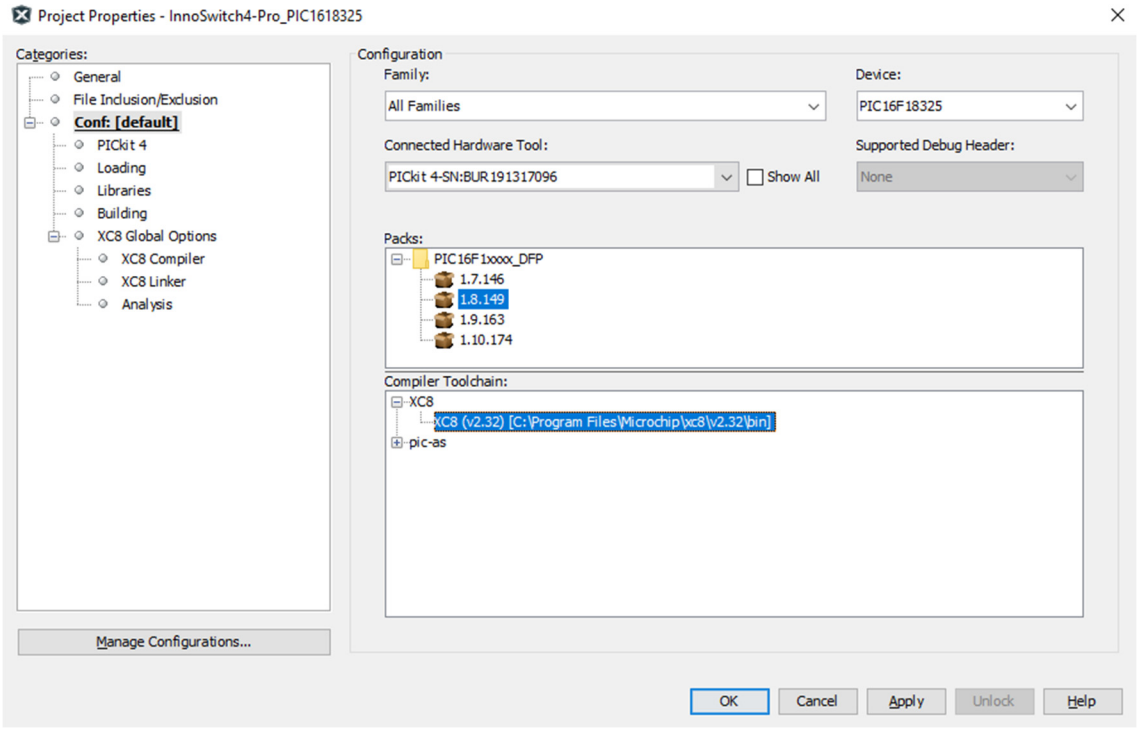


Figure 9. Project Property Settings for InnoSwitch4-Pro Library

Clean and build the project before uploading the library into the PIC microcontroller to check for errors in the configuration.

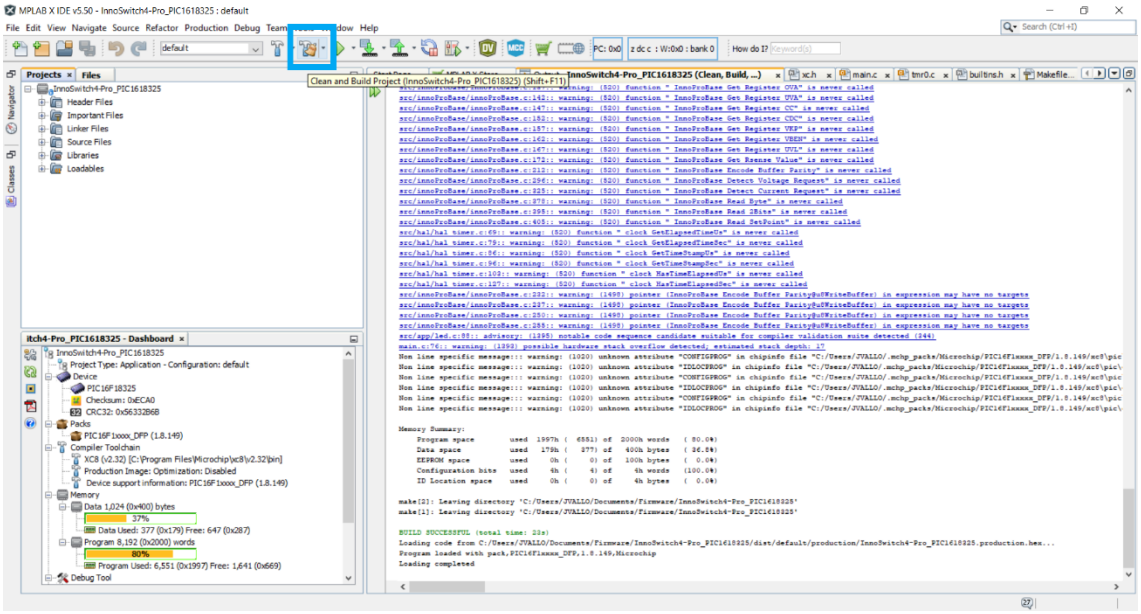


Figure 9. Clean and Build Project

Revision	Notes	Date
A	Initial release.	01/20/23

For the latest updates, visit our website: www.power.com

Power Integrations reserves the right to make changes to its products at any time to improve reliability or manufacturability. Power Integrations does not assume any liability arising from the use of any device or circuit described herein. POWER INTEGRATIONS MAKES NO WARRANTY HEREIN AND SPECIFICALLY DISCLAIMS ALL WARRANTIES INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF THIRD PARTY RIGHTS.

Patent Information

The products and applications illustrated herein (including transformer construction and circuits external to the products) may be covered by one or more U.S. and foreign patents, or potentially by pending U.S. and foreign patent applications assigned to Power Integrations. A complete list of Power Integrations patents may be found at www.power.com. Power Integrations grants its customers a license under certain patent rights as set forth at www.power.com/ip.htm.

Life Support Policy

POWER INTEGRATIONS PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF POWER INTEGRATIONS. As used herein:

1. A Life support device or system is one which, (i) is intended for surgical implant into the body, or (ii) supports or sustains life, and (iii) whose failure to perform, when properly used in accordance with instructions for use, can be reasonably expected to result in significant injury or death to the user.

2. A critical component is any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

Power Integrations Worldwide Sales Support Locations

World Headquarters
5245 Hellyer Avenue
San Jose, CA 95138, USA
Main: +1-408-414-9200
Customer Service:
Worldwide: +1-65-635-64480
Americas: +1-408-414-9621
e-mail: usasales@power.com

China (Shanghai)
Rm 2410, Charity Plaza, No. 88
North Caoxi Road
Shanghai, PRC 200030
Phone: +86-21-6354-6323
e-mail: chinasales@power.com

China (Shenzhen)
17/F, Hivac Building, No. 2, Keji Nan 8th Road, Nanshan District,
Shenzhen, China, 518057
Phone: +86-755-8672-8689
e-mail: chinasales@power.com

Germany (AC-DC/LED Sales)
Einsteinring 24
85609 Dornach/Aschheim
Germany
Tel: +49-89-5527-39100
e-mail: eurosales@power.com

Germany (Gate Driver Sales)
HellwegForum 1
59469 Ense
Germany
Tel: +49-2938-64-39990
e-mail: igbt-driver.sales@power.com

India
#1, 14th Main Road
Vasanthanagar
Bangalore-560052 India
Phone: +91-80-4113-8020
e-mail: indiasales@power.com

Italy
Via Milanese 20, 3rd. Fl.
20099 Sesto San Giovanni (MI) Italy
Phone: +39-024-550-8701
e-mail: eurosales@power.com

Japan
Yusen Shin-Yokohama 1-chome Bldg.
1-7-9, Shin-Yokohama, Kohoku-ku
Yokohama-shi,
Kanagawa 222-0033 Japan
Phone: +81-45-471-1021
e-mail: japansales@power.com

Korea
RM 602, 6FL
Korea City Air Terminal B/D, 159-6
Samsung-Dong, Kangnam-Gu,
Seoul, 135-728, Korea
Phone: +82-2-2016-6610
e-mail: koreasales@power.com

Singapore
51 Newton Road
#19-01/05 Goldhill Plaza
Singapore, 308900
Phone: +65-6358-2160
e-mail: singaporesales@power.com

Taiwan
5F, No. 318, Nei Hu Rd., Sec. 1
Nei Hu Dist.
Taipei 11493, Taiwan R.O.C.
Phone: +886-2-2659-4570
e-mail: taiwansales@power.com

UK
Building 5, Suite 21
The Westbrook Centre
Milton Road
Cambridge
CB4 1YG
Phone: +44 (0) 7823-557484
e-mail: eurosales@power.com